Sensitivity of Coverage-Based Software Reliability Models to Variations in the Operational Profile

Odair Jacinto da Silva, Marcos Lordello Chaim, Adalberto Nobiato Crespo (Co-orientador), Mario Jino (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

Abstract

Software in production may be utilized by users with diverse profiles. The way software is used affects the reliability perceived by its users; that is, software reliability may not be the same for different operational profiles. Two software reliability growth models based on structural testing coverage were evaluated with respect to their sensitivity to variations in operational profile. An experiment was performed on a real program (SPACE) with real defects, submitted to three distinct operational profiles. Testing coverage was measured according to the following criteria: all-nodes; all-arcs; and all-potential-uses. Reliability measured for each operational profile was compared to the reliabilities estimated by the two models; estimated reliabilities were obtained using the coverage for the three criteria. Results from the experiment show that the predictive ability of the two models is not affected by variations in the operational profile of the program. They also reinforce the soundness of the approach that uses code coverage information to derive software reliability models.

Keywords: software reliability, operational profile, software test, test-coverage, testing criteria.

1. Introduction

Some of the first studies and models of software reliability were proposed by Jelinski and Moranda [1] and Shooman [2]. In the eighties, research effort on software reliability became more intense and many other models appeared. Software testing time is adopted as the control variable for estimating software reliability by most of these models.

Nonetheless, several experiments and studies have shown the existence of a close relationship between code coverage and software reliability [3]. An early experiment using control-flow and data-flow based testing was conducted by Frate, Mathur and Pasquini [4]; their results provide evidence of a relationship between software reliability and coverage of elements required by the testing criteria. Another experiment, by Crespo et al. [5], investigates the relationships between software reliability and coverage of elements required by the following testing criteria: all-nodes, all-arcs, and the data-flow based family of criteria – potential-uses, all-potential-uses, all-potential-uses/du, and all-potential-du-paths [6].

The approach of reliability estimation making use of test coverage based models has been motivated from these studies; this approach has achieved better results than those from the traditional approach [7], [8].

For measuring software reliability, the inputs to the program are chosen according to the estimated probability with which they occur in production, forming the operational profile. Software in production may be utilized by users with diverse profiles. The way software is used affects the reliability perceived by its users; that is, software reliability may not be the same for different operational profiles. Hence, operational profiles are an important factor in software reliability estimation.

Many authors report results showing the close relationship between software reliability and operational profile [9], [10].

We aim to assess the robustness, with respect to different operational profiles, of two software reliability models based on code coverage. A robust model can be used to estimate precisely the software reliability for different user profiles.

The first model, called Binomial Software Reliability Model Based on Code Coverage – BMBC [7], adopts the main assumption that software contains a finite number of faults. The second model, called
Infinite Failure Software Reliability Model Based on Code Coverage – IFMBC [8], adopts the main assumption that software contains an infinite number of faults.

Three distinct operational profiles were used to evaluate the sensitivity of these models to variations in the operational profile of software; for this purpose, results from the application of the two models to estimate the reliability of a program for these operational profiles were analyzed.

2. Description of the Experiment

The application software used in the experiment was developed by the European Space Agency – ESA, in the language C; it consists of approximately 10,000 lines of code, 6,000 being lines of executable code. The system comprises a main program and 134 routines interconnected through parameters.

33 faults were revealed, corrected, and recorded during the integration testing and the operational use of the software; thus, the recorded failures in the software were caused by real faults detected in testing and in the operational use of the system. To conduct the experiment 29 faults were reinserted in the software.

In the experiment, 24, 28 and 25 faults of the 29 reinserted faults were detected, depending on the operational profile.

2.1 Operational Profiles

Operational profiles for program SPACE are defined based on its functions. A control flow graph represents the connections among the functions; each node in the graph represents a function. Two nodes are connected by an arc in the graph if control can flow between them. A path from the start node to the final node of the graph represents one execution of the program. A relative frequency value is assigned to each arc connecting two nodes. An operational profile is defined by assigning relative frequency values to all arcs of the graph.

2.2 Reliability and Coverage Measurement

The values of reliability were estimated using Nelson's method [12], that is, the “brute force” method, to establish the relationship of software reliability to other variables.

In this method, for each occurrence of a failure and after the removal of the corresponding fault, the software is executed with thousands of randomly generated test data. The ratio is calculated between "ne", the number of executions with failure, and "n", the total number of executions of the software. This ratio is an estimate of the probability of occurrence of a failure in the software. The reliability of the software is then estimated as:

\[ R = 1 - \frac{n_e}{n} \]

For each fault removed from the software, the method is applied again to recalculate the reliability. Thus, an estimate of the behavior of the reliability growth of the software is obtained as a function of fault removal. Notice that reliability is always calculated after the removal of a fault.

This procedure was done for each operational profile.

Table 1 shows the measured reliabilities for the three operational profiles. The first column of each table shows the accumulated number of removed faults; the second column shows the removed fault identification; the third column shows the number of test data between consecutive failures; the fourth column, the number of accumulated test data; and the fifth column, the measured reliability. Figure 1 shows graphically the measured reliability for each operational profile.

### Table 1 – Measured Reliability – OP1

<table>
<thead>
<tr>
<th>Removed Faults</th>
<th>Fault Identifier</th>
<th># Test Data Between Failures</th>
<th># Accumulated Test Data</th>
<th>Measured Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0.0078</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>0.0078</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>1</td>
<td>3</td>
<td>0.0102</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
<td>4</td>
<td>0.0102</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>1</td>
<td>5</td>
<td>0.0102</td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td>1</td>
<td>6</td>
<td>0.0102</td>
</tr>
<tr>
<td>7</td>
<td>24</td>
<td>1</td>
<td>7</td>
<td>0.0102</td>
</tr>
<tr>
<td>8</td>
<td>28</td>
<td>1</td>
<td>8</td>
<td>0.0464</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>2</td>
<td>10</td>
<td>0.1278</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>2</td>
<td>12</td>
<td>0.1585</td>
</tr>
<tr>
<td>11</td>
<td>19</td>
<td>2</td>
<td>14</td>
<td>0.2101</td>
</tr>
<tr>
<td>12</td>
<td>25</td>
<td>2</td>
<td>16</td>
<td>0.5147</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>6</td>
<td>22</td>
<td>0.5879</td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>7</td>
<td>29</td>
<td>0.8384</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>7</td>
<td>36</td>
<td>0.8734</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>19</td>
<td>55</td>
<td>0.8995</td>
</tr>
<tr>
<td>17</td>
<td>23</td>
<td>20</td>
<td>75</td>
<td>0.9165</td>
</tr>
<tr>
<td>18</td>
<td>17</td>
<td>31</td>
<td>106</td>
<td>0.9463</td>
</tr>
<tr>
<td>19</td>
<td>7</td>
<td>63</td>
<td>169</td>
<td>0.9710</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>63</td>
<td>232</td>
<td>0.9969</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>216</td>
<td>448</td>
<td>0.9927</td>
</tr>
<tr>
<td>22</td>
<td>27</td>
<td>243</td>
<td>691</td>
<td>0.9961</td>
</tr>
<tr>
<td>23</td>
<td>12</td>
<td>250</td>
<td>941</td>
<td>0.9961</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>250</td>
<td>1191</td>
<td>0.9987</td>
</tr>
</tbody>
</table>

Code coverage was determined using POKE-TOOL, a coverage analysis tool [13], for the following test criteria: all-nodes, all-arcs, and all-potential-uses.
2.3 Steps of the Experiment

The following steps were performed for each operational profile (OP1, OP2, and OP3):

- Test data generation: test data are generated randomly according to the relative frequencies given by the operational profile.

- Coverage measurement: for each test criterion, the correspondent coverage achieved by the generated test data is determined.

- Reliability measurement: after the occurrence of each failure in the program, the detected fault is removed and the reliability is measured using Nelson’s method.

- Reliability estimation: reliability models BMBC and IFMBC are applied to estimate reliability growth for the operational profile.

- Reliability comparison: measured reliability is compared to reliability estimated by the models, visually in the figures and statistically by the Kolmogorov-Smirnov test.

The results from the comparison between the measured reliability and the reliability estimated by the models, for the three operational profiles, would provide an indication of the robustness of the models with respect to variation in the operational profile.

3. Analysis of the Results

3.1 Results and Analysis for BMBC

Due to space restriction only results from the operational profile 1 will be showed.

Figure 2 shows the growth curves for both, the measured reliability and the estimated reliability for profile OP1. Reliability was estimated using the coverage of each test criterion. It can be seen that estimated reliability growth curves are practically identical for all the criteria, as they should be, since reliability is a software characteristic associated intrinsically to each operational profile.

It can also be seen that values estimated by the model, for the three profiles, are well fitted to the values of the measured reliability. For all profiles the hypothesis of equality of the measured reliability and estimated reliability is not rejected.

<table>
<thead>
<tr>
<th>K-S Statistics</th>
<th>α = 1%</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP1</td>
<td>0.2608</td>
<td>0.4347</td>
</tr>
</tbody>
</table>

3.2 Results and Analysis for IFMBC

Figure 3 shows the growth curves for both, the measured reliability and the estimated reliability for profile OP1. It can also be seen that estimated reliability growth curves are practically identical for all the criteria.

In this case, the values estimated by the model, for the three profiles, are also well fitted to the values of the measured reliability. For all the profiles the hypothesis of equality of the measured reliability and estimated reliability is not rejected.
4. Conclusions

We evaluated the sensitivity to variations in the operational profile of two coverage-based software reliability growth models. An experiment was carried out with program SPACE with this purpose, using three distinct operational profiles.

- The experiment has shown that models BMBC and IFMBC are sensitive to distinct operational profiles, that is, variations in the operational profile do not affect the predictive ability of the models: estimated reliabilities are very close to measured reliabilities for the three operational profiles.
- Graphical analysis (Figures 2 and 3) indicates a very good fitness of the estimated reliabilities to the measured reliabilities.
- This good fitness is also shown statistically in Tables 2 and 3.
- The desirable property of underestimation of the reliability holds for both models for the three operational profiles.
- A threat to the validity of the experiment is that it is a single program experiment; similar experiments should be performed with programs from different application domains.

5. References


